UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/726,470	12/02/2003	Shridhar Mukund	ADAPP223	5856

25920 7590 03/04/2010

MARTINE PENILLA & GENCARELLA, LLP
710 LAKEWAY DRIVE
SUITE 200
SUNNYVALE, CA 94085

EXAMINER	
MOLL, JESSE R	

ART UNIT	PAPER NUMBER
2181	

MAIL DATE	DELIVERY MODE
03/04/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/726,470	MUKUND ET AL.	
	Examiner	Art Unit	
	JESSE R. MOLL	2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 October 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Narayan et al. (U.S. Patent No. 5,822,559), herein referred to as Narayan et al.'559 in view of Kregness et al. (U.S. Patent No. 4,595,911) herein referred to as Kregness and Amagai et al. (U.S. Patent No. 7,130,312 B1) herein referred to as Amagai.

Referring to claim 1, Narayan et al.'559 discloses, as claimed, a networking application processor (see Fig. 2), comprising: an input socket (Any input to the processor is an input socket. This includes, but is not limited to: Connection from main memory to Prefetch Unit 202, Connection to main memory from Data Cache 224, clock signal, I/O requests, etc... Additionally, Narayan discloses specific paths for communication; see Fig. 67 and col. 209, lines 1-9. Finally, a processor inherently must

receive data [including a program to execute] in order to perform useful calculations) configured to receive data (the data received by the processor including instructions and data associated with the instructions); a memory (such as Main Instruction Cache 204, see Fig. 2 or reservation stations 210A-D) for storing instructions (see col. 3, lines 64-66 and col. 7, lines 36-45); circuitry (such as Instruction Alignment Unit; see Fig. 2) configured to access data structures (such as instructions or data associated with a program) associated with the received data (instruction and program data is associated with data recieved), the circuitry configured to access data structures enabling a single clock cycle access (see Fig. 20 and col. 64, lines 23-27 regarding fetching the instructions in one clock cycle) an operand from memory (such as from instruction cache in the case of an immediate operand; see col. 7, lines 11-17 regarding using immediate data and lines 23-27 regarding fetching instructions) during consecutive clock cycles (see col. 1, lines7-10 regarding the processor being a superscalar processor; *Note that a superscalar processor executes multiple instructions per clock cycle and therefore data and instructions must be read every cycle*); an arithmetic logic unit (ALU) (inside function units 212A-212D, see Fig. 2 and col. 114, lines 50-55); and circuitry for aligning operands (operand steering section; see fig. 29; and col. 113, lines 17-30 regarding aligning [extending] the A, B and D operands) to be processed by the ALU, wherein the circuitry for aligning the operand supplies an extension (zero extension or sign extension; see col. 113, lines 17-30) to the operand to allow the ALU to process different size operands (see col. 17, lines 32-42, regarding the operand size may be 8, 16, and 32 bits).

Narayan does not expressly disclose the circuitry for aligning operands causing the operands to be aligned by a lowest significant bit.

Kregness teaches the circuitry for aligning operands causing the operands to be aligned by a lowest significant bit (see col. 1, lines 30-37 regarding bit placement and fig. 19a regarding aligning multiple size operands to the right [LSB]).

At the time of the invention, it would have been obvious for one of ordinary skill in the art to have modified the system of Narayan by aligning operands by a lowest significant bit, as taught by Kregness, in order to easily perform arithmetic on the operands. Additionally, there are only a limited of ways to align data (aligning by most significant bit or least significant bit) therefore giving the modification highly predictable results.

Narayan does not expressly disclose the data being associated with a processing stage of a data packet, each processing stage of the data packet being associated with one network protocol from a plurality of network protocols in the data packet or data associated with the corresponding network protocol.

Amagai teaches data associated with a processing stage of a data packet, each processing stage of the data packet being associated with one network protocol from a plurality of network protocols in the data packet and data associated with the corresponding network protocol (see figs. 4a-4c).

At the time of the invention, it would have been obvious for one of ordinary skill in the art to have modified the system of Narayan by processing data associated with a processing stage of a data packet, each processing stage of the data packet being

associated with one network protocol from a plurality of network protocols in the data packet, as taught by Amagai, in order to facilitate inter-computer communication by using protocols that must be encoded and decoded in stages such as the OSI model (see Amagai, col. 1, lines 15-40).

Referring to claim 14, Narayan et al.'559 discloses, as claimed, a processor capable of processing a data packet (the data received by the processor such as instructions or cache lines) associated with a processing stage (fetch stage) of a pipeline of processors (see col. 1, lines 15-20; each pipeline stage does processing and therefore any pipelined processor contains a "pipeline of processors") the processor comprising: a data random access memory (RAM) (such as Main Memory; see fig. 67; or Instruction Cache 204, see Fig. 2) configured to enable access to data structures; instruction fetch and decode circuitry (comprising such as early decode units 207A-207D and MROM and anything else relating to decoding, see Fig. 2) configured to interpret instructions to be executed by an arithmetic logic unit (ALU) (inside function units 212A-212D, see Fig. 2 and col. 114, lines 50-55), the instruction fetch and decode circuitry including, a read only memory (ROM) (such as portion of the MROM of the Narayan et al.'559's system comprising the microcode sequences; such as the exception routine described on col. 145, lines 50-56; *Note that the fundamental principle of an MROM [Microcode ROM] is to hold microinstruction routines for more complicated Macroinstructions which are not easily decoded in a ROM; MROM unit access this store and outputs the microinstructions. Additionally, Narayan points out that for each*

instruction, MROM has an "entry point"; see col. 92, lines 19 and 20), the ROM configured to store code common to each processing stage associated with a pipeline of processors; a code RAM (The Section of MROM unit 209 storing MROM instructions, see Fig. 2; col. 14, lines 27-35 regarding storing MROM instructions), the code RAM configured to download code specific to the processing stage; and instruction decode circuitry (comprising such as decode units 208A-208D, see Fig. 2) configured to recognize operating instructions; execute and write back circuitry (comprising function units 212A-212D, see Fig. 2) configured to set up operands to be processed by the ALU, the execute and write back circuitry including, internal registers (inside register file 218, see Fig. 2) for defining a first and a second operand; an arithmetic logic unit (inside function units 212A-212D, see Fig. 2 and col. 114, lines 50-55) for processing the first and second operands; and align function circuitry for aligning the first and the second operands to be processed by the ALU (operand steering section; see fig. 29; and col. 113, lines 17-30 regarding aligning [extending] the A, B and D operands), wherein the align function circuitry supplies an extension ((zero extension or sign extension; see col. 113, lines 17-30) to the each of the operands to allow the ALU to transparently process different size operands (see col. 17, lines 32-42, regarding the operand size may be 8, 16, and 32 bits) wherein the execute and write back circuitry executes an instruction while the instruction fetch and decode circuitry fetches a next instruction of the instruction being executed, wherein the processor processes data associated with the processing stage (see col. 1, lines 14-17 regarding pipelining).

Narayan does not expressly disclose the circuitry for aligning operands causing the operands to be aligned by a lowest significant bit.

Kregness teaches the circuitry for aligning operands causing the operands to be aligned by a lowest significant bit (see col. 1, lines 30-37 regarding bit placement and fig. 19a regarding aligning multiple size operands to the right [LSB]).

At the time of the invention, it would have been obvious for one of ordinary skill in the art to have modified the system of Narayan by aligning operands by a lowest significant bit for the reasons stated above regarding claim 1.

Narayan does not expressly disclose the data being associated with a processing stage of a data packet, each processing stage of the data packet being associated with one network protocol from a plurality of network protocols in the data packet.

Amagai teaches data associated with a processing stage of a data packet, each processing stage of the data packet being associated with one network protocol from a plurality of network protocols in the data packet (see figs. 4a-4c).

At the time of the invention, it would have been obvious for one of ordinary skill in the art to have modified the system of Narayan by processing data associated with a processing stage of a data packet, each processing stage of the data packet being associated with one network protocol from a plurality of network protocols in the data packet, as taught by Amagai, in order to facilitate inter-computer communication by using protocols that must be encoded and decoded in stages such as the OSI model (see Amagai, col. 1, lines 15-40).

As to claim 2, Narayan et al.'559 also discloses: the networking application processor of claim 1, wherein the instructions have a width of 96 bits (see the instruction set in Fig. 1 when it comprises 12 bytes=96 bits), and wherein the single cycle access enables the data to be addressed and operated on in a single cycle in a single clock cycle without being placed into a register (see Fig. 20 regarding fetching and scanning in the same clock cycle and col. 64, lines 23-25 regarding fetching and muxing for alignment).

Note the definition of the word "operate" according to The American Heritage® Dictionary of the English Language, Fourth Edition is "to perform a function; work". Under this definition, a read (fetch) from memory is reasonably considered to be an operation.

As to claims 3 and 16, Narayan et al.'559 also discloses: the networking application processor of claim 1, wherein the different size operands are selected from the group consisting of 8 bit operands, 16 bit operands, and 32 bit operands (see col. 17, lines 32-42, regarding the operand size may be 8, 16, and 32 bits).

As to claim 4, Narayan et al.'559 also discloses: the networking application processor of claim 1, further including: an output socket for transmitting processed data; and a 64 bit bus (see such as 64-bit input bus to decode unit 0-3, see Fig. 25) connecting the input socket and the output socket.

As to claims 5 and 15, Narayan et al.'559 also discloses: the networking application processor of claim 1, wherein the extension to the operand fills each higher bit with a value (such as 0 for each higher bit for the unsigned operands).

As to claim 6, Narayan et al.'559 also discloses: the networking application processor of claim 1, wherein the operand is selected from the group consisting of a

source operand, a destination operand, an immediate operand, and an internal register operand (see col. 2, lines 5-23, and Fig. 1, regarding the instruction set format including source operand, a destination operand, an immediate operand).

As to claim 17, Narayan et al.'559 also discloses: the processor of claim 14, wherein the operating instructions wherein the operating instructions are formatted as 96 bit instructions (see the instruction set 100 in Fig. 1 when it comprises 12 bytes=96 bits), each of the 96 bit instructions including a single return bit (the bit such as end of file or record in the instruction set 100 in Fig. 1).

As to claim 18, Narayan et al.'559 also discloses: the processor of claim 14, wherein the processor is configured as a two stage pipeline for pipelining an instruction fetch and decode operation (using prefetch/predecode unit 202; and decode units 208A-208D, see Fig. 2) and an execute and write back operation (see Col. 144, lines 45-65, regarding write back operations).

As to claim 19, Narayan et al.'559 also discloses: the processor of claim 14, wherein the operating instructions include microcode configured to predict a likely direction for a branch instruction (using branch prediction unit 220, see Fig. 2).

As to claim 20, Narayan et al.'559 also discloses: the processor of claim 19, wherein no operation (NOP's) instructions are included (see such as col. 139, lines 4-5, regarding some of the instructions may be NOP), the NOP's configured block an invalidated pre-fetched instruction.

Referring to claim 7, Narayan et al.'559 discloses, as claimed, a processor (see Fig. 2), comprising: an input socket (the input data from I/O module of the system intended to be used) configured to receive data packets; a memory (such as main memory of the Narayan et al.'559's system or instruction cache 204, see Fig. 2) for storing instructions; circuitry configured to access data structures associated with a processing stage, the circuitry configured to access data structures enabling a single cycle access from a memory location (such as main memory of the Narayan et al.'559's system or data cache 224, see Fig. 2); and an arithmetic logic unit (ALU) (inside function units 212A-212D, see Fig. 2), the ALU configured to receive a first and a second operand (operand A and operand B, see Fig. 33); the second operand being specified from an internal register (REGF, see Fig. 33).

Narayan et al.'559 discloses the claimed invention except for explicitly showing the first operand having a mask enabling the ALU to process a non-masked segment of the first operand.

However, Narayan et al.'559 shows using masks to select bytes before sending to the register file for the further use (see Col. 144, lines 59-60).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to modify Narayan et al.'559's system to comprise the first operand having a mask enabling the ALU to process a non-masked segment of the first operand, as also taught by Narayan et al.'559, in order to facilitate selecting the useful bytes to be processed and saving the processing time in the Narayan et al.'559's system.

As to claim 8, Narayan et al.'559 also discloses: the instructions have a width of 96 bits (see the instruction set in Fig. 1 when it comprises 12 bytes=96 bits) as set forth in claim 2 above.

As to claim 9, Narayan et al.'559 also discloses: the processor of claim 7, wherein each of the instructions include a loadback feature enables random accesses to one of a source indirect register or a destination indirect register through indirect addressing (see Fig. 1 and col. 2, lines 5-22 regarding such as adding the displacement value to the content of a register to form a memory location).

As to claim 10, Narayan et al.'559 also discloses: the processor of claim 7, wherein the mask is associated with an immediate value (see Fig. 1, the instruction set comprising the immediate field) of the first operand.

As to claim 11, Narayan et al.'559 also discloses: first and the second operands are associated with a size selected from the group consisting of 8 bit operands, 16 bit operands, and 32 bit operands (see col. 17, lines 32-42, regarding the operand size may be 8, 16, and 32 bits) as set forth in claim 3 above.

As to claim 12, Narayan et al.'559 also discloses: the processor of claim 7, wherein the first operand is selected from the group consisting of a source operand, a destination operand, an immediate operand, and an internal register operand (see col. 2, lines 5-23, and Fig. 1, regarding the instruction set format including source operand, a destination operand, an immediate operand) as set forth in claim 6 above.

As to claim 13, Narayan et al.'559 also discloses: the method of claim 7, wherein the memory location (such as main memory of the Narayan et al.'559's system or data

cache 224, see Fig. 2) is a static random access memory (SRAM).

3. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Narayan et al. (U.S. Patent No. 5,822,559), herein referred to as Narayan et al.'559 in view of Kregness et al. (U.S. Patent No. 4,595,911) herein referred to as Kregness and Amagai et al. (U.S. Patent No. 7,130,312 B1) herein referred to as Amagai and Cheah (U.S. Patent No. 6,243,291).

Regarding claim 21, Narayan does not expressly teach an instruction fetch and decode operation, and an execute and writeback operation, wherein the execute and write back operation is executed simultaneously with the fetch and decode operation for a next instruction following the execute and write back operation.

Cheah teaches an instruction fetch and decode operation (Input and Address decoding; see fig. 4 and fig. 5), and an execute and writeback operation (Word address switching and Output; see fig. 5), wherein the execute and write back operation is executed simultaneously with the fetch and decode operation for a next instruction following the execute and write back operation (see fig. 5).

At the time of the invention, it would have been obvious for one of ordinary skill in the art to have modified the invention of Narayan by using a 2-stage pipeline, as taught by Cheah, in order to simplify the processor and reduce power consumption.

Response to Arguments

4. Applicant's arguments with respect to aligning operand to a lowest significant bit have been considered but are moot in view of the new ground(s) of rejection.

5. Applicant's arguments filed 30 October 2009 have been fully considered but they are not persuasive. See the rejection above for a detailed analysis of the limitations and new grounds of rejection.

6. Regarding the argument that Narayan does not teach concurrent operations every clock cycle, Examiner respectfully disagrees. In any superscalar processor, instructions are executed every cycle. The instructions must be pulled from a memory (such as the cache, reservation stations, inter-stage registers, etc...) each clock cycle. Additionally, using fast (single cycle) memory within a high performance processor is well known in the art and it would have been obvious to add to a processor

7. Regarding the argument that Narayan does not teach a mask within the instruction. The mask is considered to be part of the instruction since it is used to instruct the processor how to process the data. Additionally, masking writes of instructions is well known in the art and would be obvious to add to a processor.

8. Applicant is reminded that the Examiner is available via telephone at the number and hours listed below if there are any questions or comments about the action.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

9.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to JESSE R. MOLL whose telephone number is (571)272-2703. The examiner can normally be reached on M-F 10:00 am - 6:30 pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Alford Kindred can be reached on (571)272-4037. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Alford W. Kindred/
Supervisory Patent Examiner, Art Unit 2181

Jesse R Moll
Examiner
Art Unit 2181

/J. R. M./
Examiner, Art Unit 2181